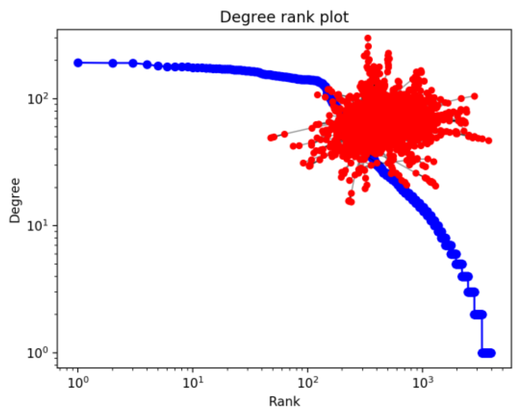
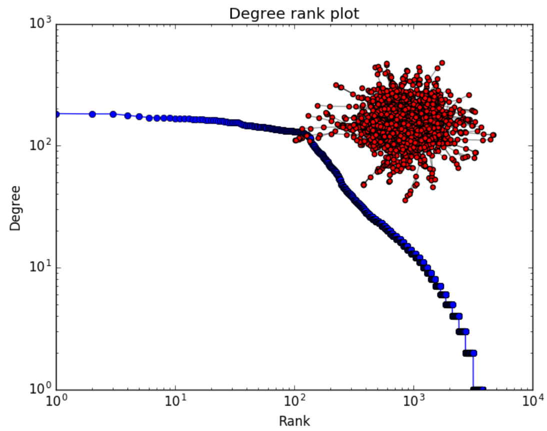
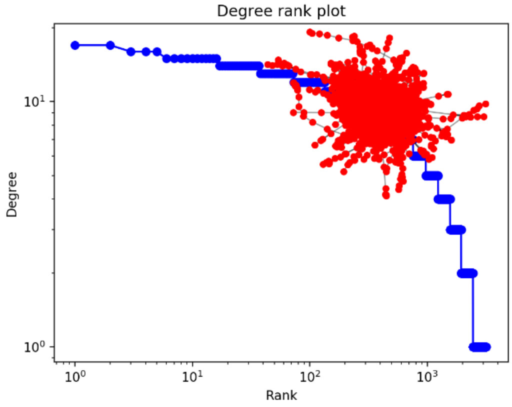
Implementation

We implemented two theoretical models for protein interaction networks (Partial Duplication model and Duplication Divergence model) and a Erdos Renyi Graph (a random interaction network). To be able to compare the graphs, we made around 3000 nodes like in the Yeast Protein Interaction Network (PIN). However, the partial duplication model was slow therefore, we only had roughly 1000 nodes for it. We (then) performed random attacks on each of the four networks. That is, we picked a random vertex and removed it from the network. Lastly, we removed vertices in decreasing order of degree for each of the network. And for both experiments we made sure that only those nodes are included that have a degree 2 or more. So, in the log-log plots you can only see the nodes with degree higher than one, because if we have nodes with degree less than two, we get a “division by zero”-error in the average shortest path function.
In our script, we have defined a function for every PIN; Yeast, Partial Duplication model, Duplication Divergence and Erdos Renyi. We also have a function that makes the experiments where we find the average shortest path length, clustering coefficient and the degree distribution. We used the probabilities $p = 0.99$, $q = 0.3$, $r = 0.6$. p is high because we wanted to have a high probability that the model will create an edge such that we can have more connected components.

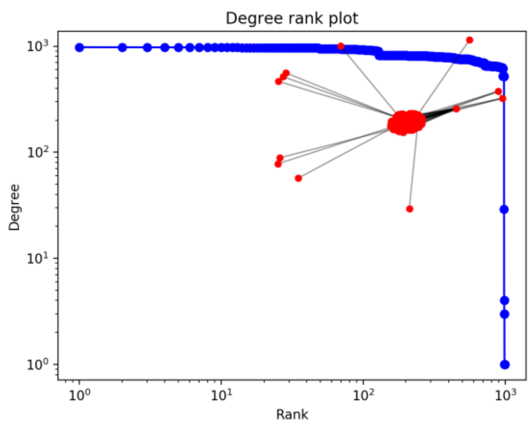
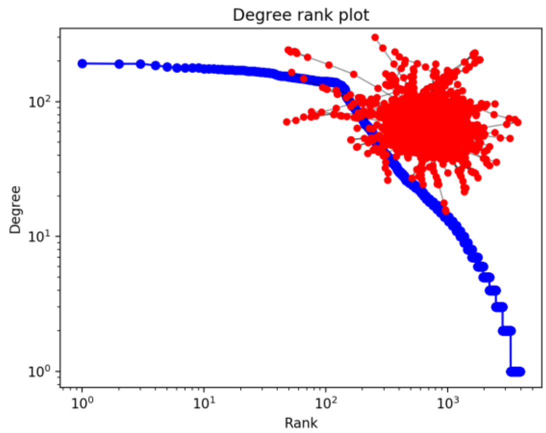
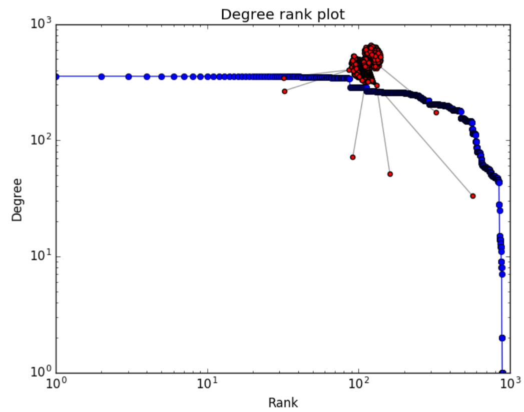
Libraries Used: Networkx, matplotlib.pyplot, random, operator

The program is in the python file pin.py and can run by just compiling it.
For every attack on the PINs, we save the loglog-plot of the degree distribution. In the tables below you can only see one of these plots, but all the other plots are in the folder: graphs.

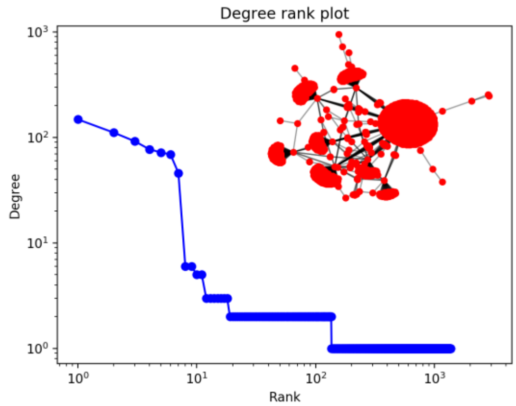
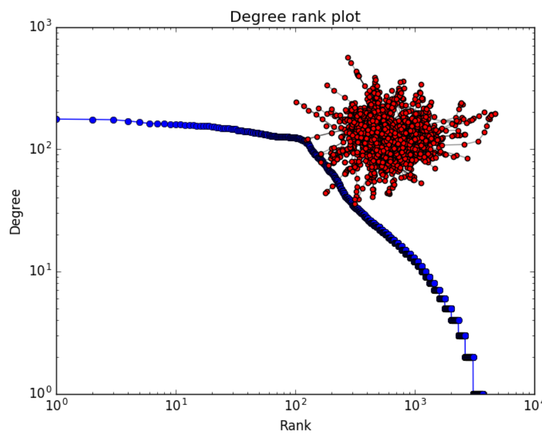
<table>
<tr><th>Yeast</th><th>Average shortest path length</th><th>Clustering coefficient</th><th>Degree distribution, loglog-plot</th></tr>
<tr><td>Without attacks</td><td>1.1650749340681341</td><td>0.4687669880389289</td><td>
</td></tr>
<tr><td colspan="4">Random attack</td></tr>
<tr><td>Remove 100 nodes</td><td>1.166297697082525</td><td>0.4648772270964038</td><td rowspan="5">
</td></tr>
<tr><td>Remove 200 nodes</td><td>1.1643699370198017</td><td>0.46312234538478314</td></tr>
<tr><td>Remove 300 nodes</td><td>1.1595462323594588</td><td>0.46195040413308497</td></tr>
<tr><td>Remove 400 nodes</td><td>1.1605841417997762</td><td>0.4571589679915093</td></tr>
<tr><td>Remove 500 nodes</td><td>1.156410570021468</td><td>0.4555473954879148</td></tr>
<tr><td colspan="4">Remove highest degree nodes (decreasing order)</td></tr>
<tr><td>Remove 500 nodes</td><td>1.1780259156991701</td><td>0.4319984298790373</td><td rowspan="3"></td></tr>
<tr><td>Remove 1000 nodes</td><td>1.212221893327937</td><td>0.39676700314866986</td></tr>
<tr><td>Remove 1500 nodes</td><td>1.280996425586471</td><td>0.3513576579191499</td></tr>
</table>

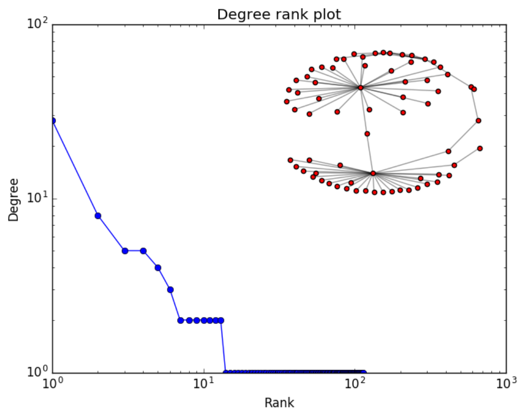
Remove 2000 nodes	1.498780619929638	0.2883799830364718	
Remove 2500 Nodes	1.313855823206495	0.20864461045891136	

The average shortest path does not change significantly when we remove random nodes, but it increases when we remove nodes with the highest degree. The clustering coefficient decreases when we remove the highest degree nodes.

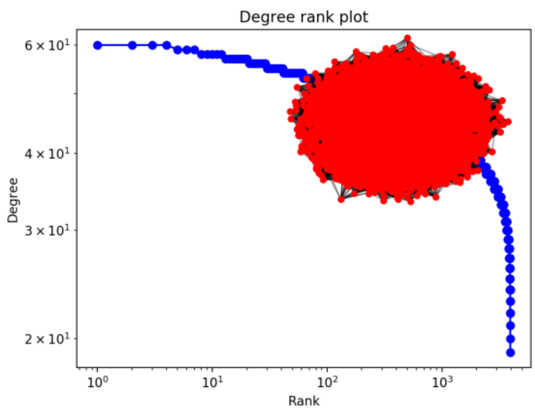
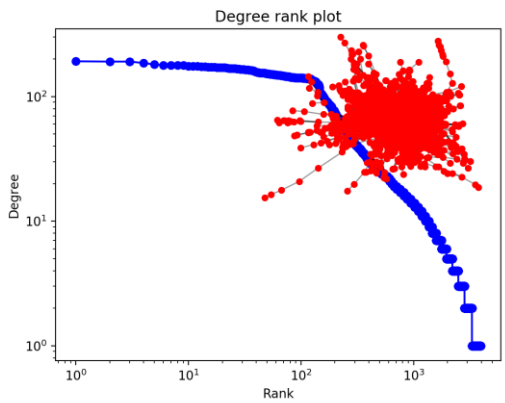
Partial Duplication Model	Average shortest path length	Clustering coefficient	Degree distribution, loglog-plot
Without attacks	1.2524654472656684	0.7739798815971932	
		Random attack	
Remove 100	1.59146391313	0.828310380405	
Remove 200	1.29413062805	0.835402465867	
Remove 300	1.30326398852	0.833711082424	
Remove 400	1.29823308292	0.834645671399	
Remove 500	1.5873987976	0.836101964913	
		Remove highest degree nodes (decreasing order)	
Remove 300	1.17489682504	0.888018045852	
Remove 600	1.1507926825	0.899527031642	
Remove 900	1.1356609784	0.871985035277	
Remove 1200	1.15715354532	0.85277240518	
Remove 1500	1.21133415419	0.756768719871	

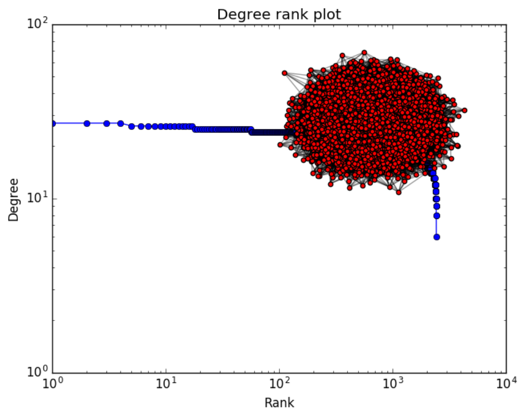
For the random attacks on the Partial Duplication model we see that the average shortest path lengths are not following a increasing or decreasing pattern, but when we remove the highest degree nodes the average path length is first decreasing and then starts to increase.

Duplication Divergence Model	Average shortest path length	Clustering coefficient	Degree distribution, loglog-plot
Without attacks	1.4225971310130145	0.01099394136680896	
Random attack			
Remove 100 nodes	1.50319448808	0.0258012885776	
Remove 200 nodes	1.44488420715	0.0220344755196	
Remove 300 nodes	1.4885104716	0.0211176306717	
Remove 400 nodes	1.48117423218	0.0138561076938	
Remove 500 nodes	1.51513540934	0.0131310640702	
Remove highest degree nodes (decreasing order)			
Remove 1 nodes	1.20815577745	0.00177510636672	
Remove 2 nodes	1.29036252182	0	

Remove 3 nodes	1.27355024257	0	
Remove 4 nodes	1.21525784157	0	
Remove 5 nodes	1.18400673401	0	

The average path length is increasing and then decreasing when we remove nodes with highest degree distribution. The clustering coefficient is very small because the Duplication Divergent model makes a PIN with few nodes that have a very high degree. When we just remove 2 nodes, the clustering coefficient drops to zero. Random attacks do not have the same effect (unless we randomly choose to remove a high clustering node)

Erdos-Renyi random graph	Average shortest path length	Clustering coefficient	Degree distribution, loglog-plot
Without attacks	2.6566853282740506	0.010062743353953116	
		Random attack	
Remove 100 nodes	2.6635818716557433	0.009967634756125181	
Remove 200 nodes	2.6703548404786948	0.00994127550328272	
Remove 300 nodes	2.6775209694173494	0.009956855540958365	
Remove 400 nodes	2.684425371957521	0.009985173894729298	
Remove 500 nodes	2.6916108741588762	0.010001217018774829	
		Remove highest degree nodes (decreasing order)	
Remove 500 nodes	2.73683470807	0.00887783042075	
Remove 1000 nodes	2.8077020621	0.00828204816499	
Remove 1500 nodes	2.93379377824	0.00760192443323	

Remove 2000 nodes	3.22196987789	0.00661461449604	
Remove 2500 nodes	3.70861200295	0.00546094172929	

We had to remove 500 nodes with highest degree at a time to see an effect on the average shortest path length which increases significantly. The clustering coefficient does not change much. The random attacks do not have the same effect.